

**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-20. (Canceled)

21. (Currently Amended) A method for managing and transmitting events from a server via a communication link to at least one client ~~where, said method comprising:~~

logging possible events ~~are-logged~~ in a client event service for the purpose of initializing ~~and/or~~ or updating the client,

logging possible events ~~are-logged~~ in a server event service for the purpose of initializing ~~and/or~~ or updating the server,

transferring detected events which have been logged ~~are-transferred~~ from an installation interface to the server event service,

sending requests initiated by the client event service regarding the detected events ~~are-made~~ to the server event service,

transmitting ~~on the basis of a request which has been made to the server event-service~~ the detected events ~~are-transmitted~~ to the client event service on the basis of a request which has been made to the server event service, and

transmitting ~~and~~ events received by the client event service ~~are-transmitted~~ to a client application,

wherein the client application logs a client callback function in the client event service for every event about which it is to be notified, and the client event service uses the communication link to log a corresponding server callback function in the server event service, and

wherein to log the callback functions for an event with which ~~the~~ a same event name is associated with the client and with the server ~~in preparation for the method~~, the following steps are performed:

calling, by the client application, ~~calls~~ a client logging function from the client event service and ~~provides~~ providing said function with ~~the~~ a name of the ~~an~~ event in question and with a pointer to the client callback function which is to be logged,

logging, by the client logging function, ~~generates~~ a unique event identifier, ~~and~~ transmits

transmitting the event identifier and the event name via the communication link to a server logging function of the server event service,

logging, by the server logging function, ~~logs~~ a server callback function with the installation interface by transferring the event name,

storing, by the server logging function in a server event table, ~~stores~~ a data record, which contains at least the event identifier and a pointer to the server callback function which is to be logged, ~~in a server event table~~,

reporting, by the server logging function, ~~reports~~ the performance of the logging operation to the client logging function of the client event service via the communication link, and

logging, by the client logging function, ~~logs~~ the client callback function by storing a data record in a client event table, the data record containing at least the event identifier and a pointer to the client callback function which is to be logged.

22. (Previously Presented) The method as claimed in claim 21, wherein the events to be transmitted are detected by a data capture unit in a technical installation and are reported to the installation interface of the server.

23. (Canceled)

24. (Previously Presented) The method as claimed in claim 21, wherein after a client callback function has been logged for the first time the client logging function starts a request generator which then makes requests for event transmission to the server event service.

25. (Previously Presented) The method as claimed in claim 24, wherein the request generator of the client event service makes the requests for event transmission to the server event service cyclically.

26. (Currently Amended) The method as claimed in claim 24, wherein events are transmitted by performing the following steps:

detecting, by the installation interface, ~~detects~~ an event which has occurred and ~~calls~~ calling the server callback function logged for this event,

producing, by the server callback function, ~~produces~~ an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission ~~the server event service reads the entry produced from the event queue and,~~

transmitting ~~transmits it~~ the entry via the communication link to the client event service ~~the client event service takes,~~

receiving, by the client event service, the entry ~~received and ascertains,~~

ascertaining and ~~calls~~ calling the client callback function logged for this the event, and

executing, by the client callback function, ~~executes~~ a defined action for the corresponding event in the client application.

27. (Currently Amended) A system for managing and transmitting events from a server via a communication link to at least one client, said system comprising:

at least one client, comprising:

at least one client event service, ~~where for the purpose of logging possible events, the client has at least one client event service~~ which uses a communication link to make requests for event transmission to a server event service,

a server, comprising:

at least one server event service, which has at least one server logging function for logging server callback functions, and for logging possible events, and which uses a communication link to transmit events to the client event service,

at least one server event table for holding data records which describe a respective logging operation, which server event table is ~~in the form of~~ formed as a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function which is to be logged, ~~and~~

at least one event queue for holding entries which describe a respective event, ~~and to transmit for transmitting received events to a client application, for the purpose of logging possible events the server has at least one server event service which uses a communication link to transmit events to a client event service, the server has, and~~

at least one installation interface which transfers events which have occurred to the at least one server event service.

28. (Currently Amended) The method as claimed in claim 21, wherein optionally a tidying function of the server event service is called which deletes the server event table and ~~the~~ an event queue if the client event service is no longer communicating with the server event service.

29. (Previously Presented) The system as claimed in claim 27, wherein the installation interface is connected to a data capture unit of a technical installation in order to read in events detected by the data capture unit.

30. (Previously Presented) The system as claimed in claim 27, wherein the server event service has at least one server callback function which can be logged for at least one event and which is called when an event for which it is logged occurs.

31. (Previously Presented) The system as claimed in claim 27, wherein the server event service has, for every client event service with which it communicates via a communication link, a separate client data record which respectively contains at least one server event table and at least one event queue.

32. (Previously Presented) The system as claimed in claim 31, wherein the server event service has a tidying function which deletes the client data record if the associated client event service is no longer communicating with the server event service.

33. (Previously Presented) The system as claimed in claim 31, wherein the server event table is in the form of a hash table and holds data records which contain at

least one event identifier and a pointer to a server callback function which is to be logged.

34. (Previously Presented) The system as claimed in claim 27, wherein the client event service has at least one client logging function for logging client callback functions, at least one client event table for holding data records which describe the log, and at least one request generator for making cyclic requests for event transmission.

35. (Previously Presented) The system as claimed in claim 34, wherein the client event table is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a client callback function which is to be logged.

36. (Currently Amended) The method as claimed in claim 21, wherein events are transmitted by performing the following steps:

detecting, by the installation interface, ~~detects~~ an event which has occurred and ~~calls~~ calling the server callback function logged for this event,

producing, by the server callback function, ~~produces~~ an entry describing the event in at least one event queue,

reading, by the server event service, the entry produced in the event queue upon the next request from the client event service for event transmission ~~the server event service reads the entry produced from the event queue and,~~

transmitting ~~transmits it~~ the entry via the communication link to ~~the client event service,~~ the client event service takes,

receiving, by the client event service, the entry ~~received and ascertains,~~  
ascertaining and ~~calls~~ calling the client callback function logged for this the  
event, and  
executing, by the client callback function, ~~executes~~ a defined action for the  
corresponding event in the client application.